

# 7º CAPÍTULO

## Análise de Movimento



Prof. Arnaldo Abrantes / anotado por Prof. Nuno Pinho da Silva

- Análise de movimento em vídeo é importante – permite revelar características (*features*) úteis, como a velocidade dos objectos ou do observador, a forma dos objectos, as suas trajectórias
  - Ao observar as alterações na intensidade (ou cor) dos pixels é possível detectar a presença de objectos e fazer o seu reconhecimento
- A detecção de alguns eventos (exemplo, transições de cena) em sequências longas de vídeo permite o seu fraccionamento (em diversos *clips*) tornando mais fácil operações como o acesso, a análise e a edição
- Quatro situações onde ocorre movimento (alterações nos pixels):
  - Câmara fixa, um só objecto em movimento, fundo constante
    - Situação simples de vigilância
  - Câmara fixa, vários objectos em movimento, fundo constante
    - Situação realista num contexto de vigilância
  - Câmara em movimento, cena relativamente imutável
    - Permite obter maior número de observações da cena, construir mosaicos, determinar profundidade relativa dos objectos, detectar eventos como *pan* ou *zoom* de câmara
  - Câmara em movimento, vários objectos em movimento
    - Veículo robótico a circular num ambiente de tráfego intenso

# Métodos de subtracção de imagem

- Subtracção de imagens consecutivas

$$|I_n(r, c) - I_{n-1}(r, c)| > T_n(r, c)$$

- Subtração de fundo

$$|I_n(r, c) - B_n(r, c)| > T_n(r, c)$$

$$B_{n+1}(r, c) = \begin{cases} B_n(r, c) & \text{se } (r, c) \text{ é pixel activo} \\ \alpha B_n(r, c) + (1 - \alpha) I_n(r, c) & \text{caso contrário} \end{cases}$$



- **Entrada:** duas imagens monocromáticas  $I_n(r, c)$  e  $I_{n-k}(r, c)$  e o limiar  $\tau$
- **Saída:** imagem binária,  $I_{out}$  e conjunto de caixas,  $B$ , com a localização dos objectos detectados

- Algoritmo com 5 passos:

1. Calcular imagem binária (pixels activos)

$$I_{out}(r, c) = \begin{cases} 1 & \text{se } |I_n(r, c) - I_{n-k}(r, c)| > \tau \\ 0 & \text{caso contrário} \end{cases}$$

2. Realizar extracção de componentes conexos sobre  $I_{out}$

3. Remover as regiões com pequena área (ruído)

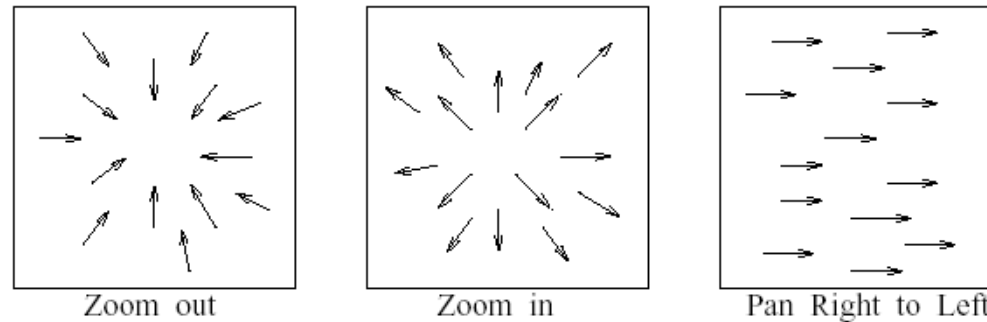
4. Realizar operação morfológica de fecho usando um pequeno disco

5. Para cada região, determinar a caixa rectangular que a contém (*bounding box*)

# Cálculo de vectores de movimento

---

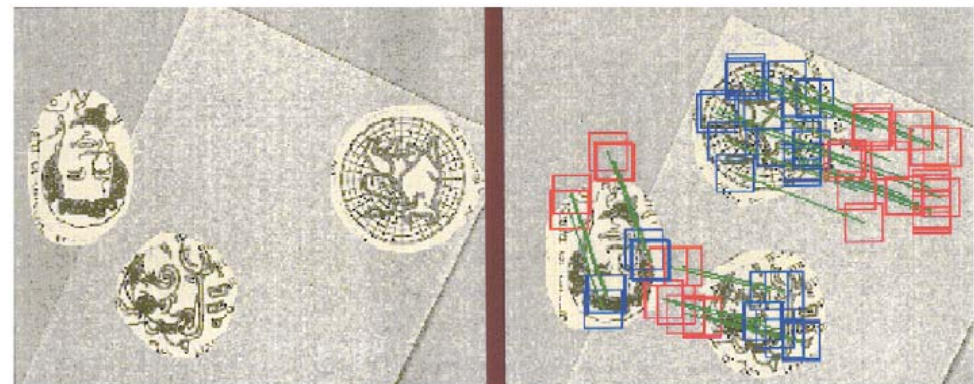
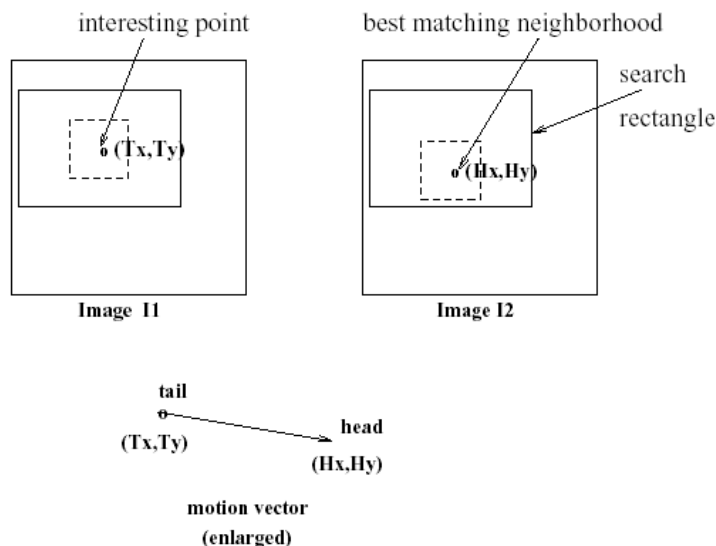
- Um conjunto de vectores 2D representando o movimento aparente (na imagem) de objectos 3D é chamado um **campo de movimento**



- O foco de expansão (**FOE**) é o ponto da imagem de onde todos os vectores divergem. O foco de contracção (**FOC**) é para onde todos os vectores convergem
- Fluxo óptico** é o campo de movimento calculado sob a hipótese de que as intensidades em redor de pontos em correspondência nas duas imagens se mantêm constante

# Método esparsa – correspondências de pontos

- Detectar pontos de interesse na primeira imagem (por exemplo, algoritmo 9.2 na Shapiro, página 258, para detecção de cantos)
- Por cada ponto de interesse detectado na primeira imagem ( $T_x, T_y$ ) procurar na segunda imagem o ponto correspondente (por exemplo, método de correlação)
- Caso o *match* seja bom, as coordenadas do ponto definem a cabeça do vector movimento ( $H_x, H_y$ )



**Find interesting points of a given input image.**

```
procedure detect_corner_points(I, V);  
{  
    “I[r, c] is an input image of MaxRow rows and MaxCol columns”  
    “V is an output set of interesting points from I.”  
    “ $\tau$  is a threshold on the interest operator output”  
    “ $w$  is the halfwidth of the neighborhood for the interest operator”  
    for r := 0 to MaxRow - 1  
        for c := 0 to MaxCol - 1  
            {  
                if I[r, c] is a border pixel then break;  
                else if ( interest_operator (I, r, c, w )  $\geq \tau_1$  ) then add [(r, c), (r, c)] to set V;  
                “The second (r, c) is a place holder in case vector tip found later.”  
            }  
}  
real procedure interest_operator (I, r, c, w )  
{  
    “ $w$  is the halfwidth of operator window”  
    “See alternate texture-based interest operator in the exercises.”  
    v1 := variance of intensity of horizontal pixels I1[r, c - w] ... I1[r, c + w];  
    v2 := variance of intensity of vertical pixels I1[r - w, c] ... I1[r + w, c];  
    v3 := variance of intensity of diagonal pixels I1[r - w, c - w] ... I1[r + w, c + w];  
    v4 := variance of intensity of diagonal pixels I1[r - w, c + w] ... I1[r + w, c - w];  
    return minimum {v1, v2, v3, v4};  
}
```

Ver también SIFT  
(Learning OpenCV,  
Invariant Features, pg.  
321)

**Algorithm 17:** Algorithm for detecting interesting image points.



# Algoritmo análogo – compressão MPEG

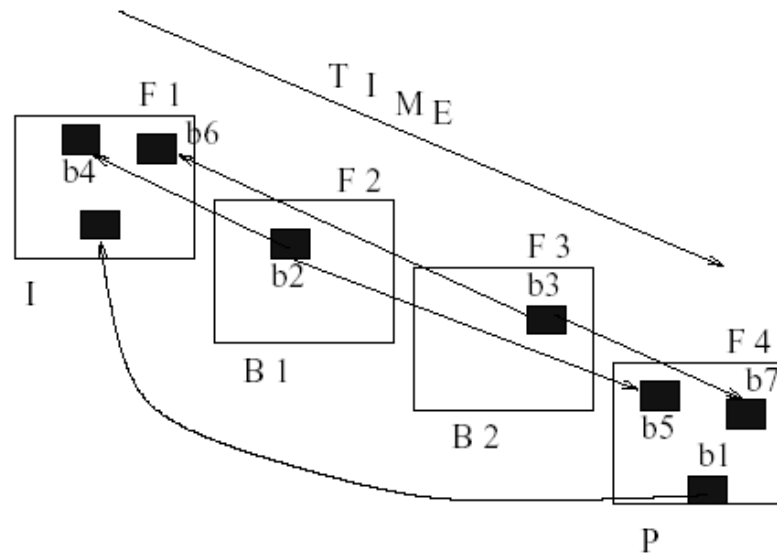


Figure 9.8: A coarse view of the MPEG use of motion vectors to compress the video sequence of four frames F1, F2, F3, F4. F1 is coded as an *independent* (I) frame using the JPEG scheme for single still images. F4 is a P frame *predicted* from F1 using motion vectors together with block differences: 16 x 16 pixel blocks (b1) are located in frame F1 using a motion vector and a block of differences to be added. *Between frames* B1 and B2 are determined entirely by interpolation using motion vectors: 16 x 16 blocks (b2) are reconstructed as an average of blocks (b4) in frame F1 and (b5) in frame F4. Between frames F2 and F3 can only be decoded after predicted frame F4 has been decoded even though these images were originally created before F4. Between frames yield the most compression since each 16 x 16 pixel block is represented by only two motion vectors. I frames yield the least compression.



# Método denso – fluxo óptico

- Representação da imagem (modelo contínuo) usando a série de Taylor (1ª ordem)

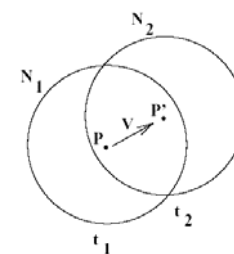
$$f(x + \Delta x, y + \Delta y, t + \Delta t) = f(x, y, t) + \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y + \frac{\partial f}{\partial t} \Delta t + h.o.t.$$

- Hipótese de intensidade constante

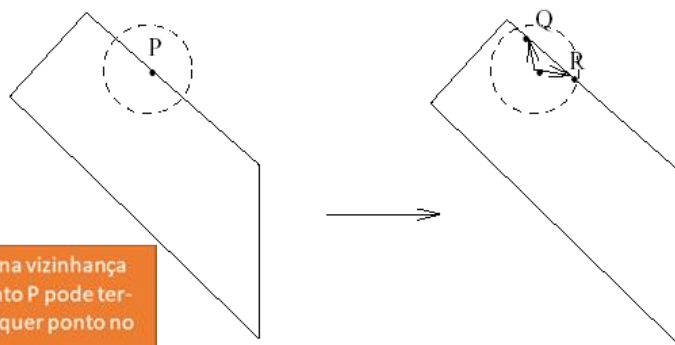
$$f(x + \Delta x, y + \Delta y, t + \Delta t) = f(x, y, t) \longrightarrow$$

- Resultado:

$$-\frac{\partial f}{\partial t} \Delta t = \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) \circ (\Delta x, \Delta y)$$

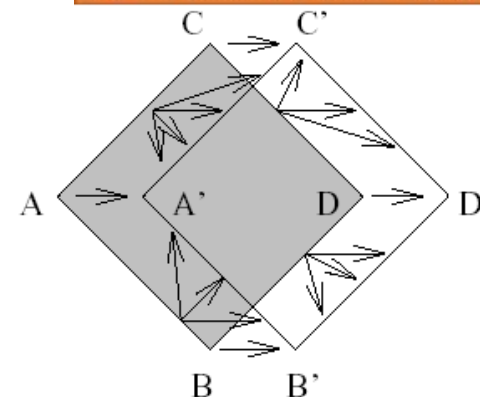


Dada a natureza dos cantos, o fluxo óptico nestes pontos de interesse podem ser calculados com alguma precisão e aberturas pequenas.



Problema de abertura: Devido à pequena vizinhança utilizada para a correspondência, o ponto P pode ter-se movido para Q, para R, ou para qualquer ponto no segmento QR.

Problema de abertura



Importância dos cantos

## Algoritmo rápido (Freeman *et. al.*)

---

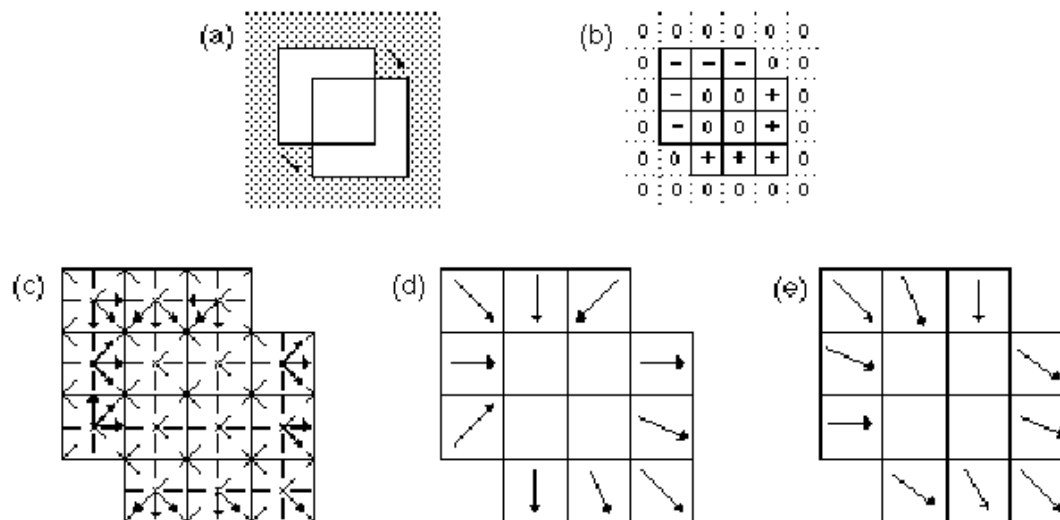
- Calcular a diferença, entre a imagem actual e a anterior

$$d_n(r, c) = I_n(r, c) - I_{n-1}(r, c)$$

- Para todos os pixels onde a diferença seja não nula (ou não muita pequena), enumerar todos os possíveis movimentos consistentes com as observações:
  1. Se  $d_n(r, c) < 0$  então é porque o movimento se faz em direcção ao vizinho com maior intensidade
  2. Se  $d_n(r, c) > 0$  então é porque o movimento se faz em direcção ao vizinho com menor intensidade
- Considerar, no passo anterior, quatro direcções possíveis (quatro tipos de vizinhança): horizontal, vertical e as duas diagonais
- Tratar cada uma das 4 estimativas como vectores e realizar a sua soma, para cada pixel
- Finalmente, estimar o fluxo óptico em cada pixel como a média espacial, calculada numa vizinhança 3x3

# Ilustração do algoritmo rápido

	(1)	(2)	(3)	(4)
<u>edge contrast</u> <u>and its motion</u> <u>direction</u>				
<u>pixel value</u> <u>subtraction</u> (positive or negative)				



- Objectivo: Detecção de alterações significativas em vídeo
  - Mudança de cena
  - Mudança de shot
  - Pan
  - Zoom in e zoom out
  - Efeitos especiais

- fade in, fade out

$$C = A\left(1 - \frac{t}{T}\right)u(T - t) + B\frac{t - T}{T}u(t - T)$$

- Dissolve

$$C = A\left(1 - \frac{t}{T}\right) + B\frac{t}{T}$$

- wipe



- Solução simples (e estúpida)

$$d(I_1, I_2) = \frac{\sum_{r=0}^{N-1} \sum_{c=0}^{M-1} |I_1(r, c) - I_2(r, c)|}{NM}$$

- Melhor solução: partir a imagem em blocos rectangulares e calcular médias e variâncias em cada bloco

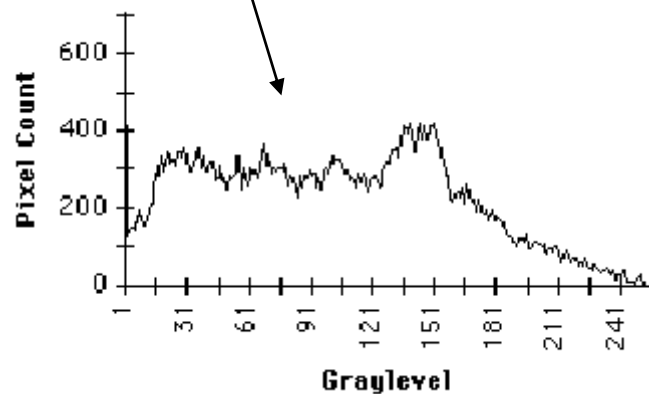
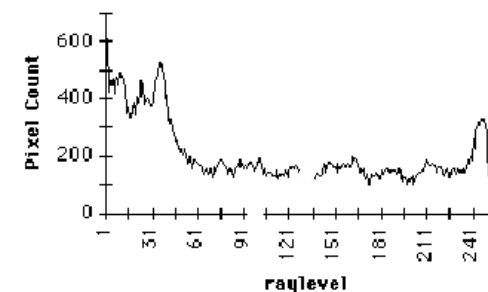
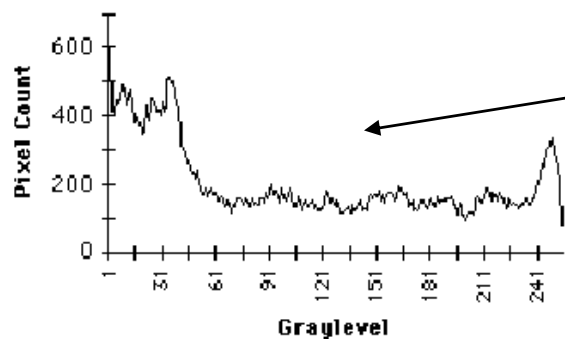
$$d_{block}(B_i, B_j) = \begin{cases} 1 & \text{se } r > \tau \\ 0 & \text{se } r \leq \tau \end{cases} \quad r = \frac{\left[ \frac{\sigma_i^2 + \sigma_j^2}{2} + \left( \frac{\mu_i - \mu_j}{2} \right)^2 \right]^2}{\sigma_i^2 \sigma_j^2}$$

Verosimilhança  
(Likelihood ratio)

$$d(I_1, I_2) = \sum_{B_{1i} \in I_1; B_{2i} \in I_2} d_{block}(B_{1i}, B_{2i})$$

Alternativa – distâncias entre histogramas

# Distâncias entre histogramas



Histograma de 64 bins de imagens RGB: level \in {0,63}: concatenar os 2 bits de maior ordem dos três canais de cor num só bin.

Discuta as vantagens/limitações em relação aos métodos anteriores, nomeadamente referindo-se à

- velocidade de aplicação,
- Coerência da distribuição espacial?

- Fade out seguido de fade in (duração  $2T$ )

$$C = A\left(1 - \frac{t}{T}\right)u(T - t) + B\frac{t - T}{T}u(t - T)$$

- Dissolve

$$C = A\left(1 - \frac{t}{T}\right) + B\frac{t}{T}$$

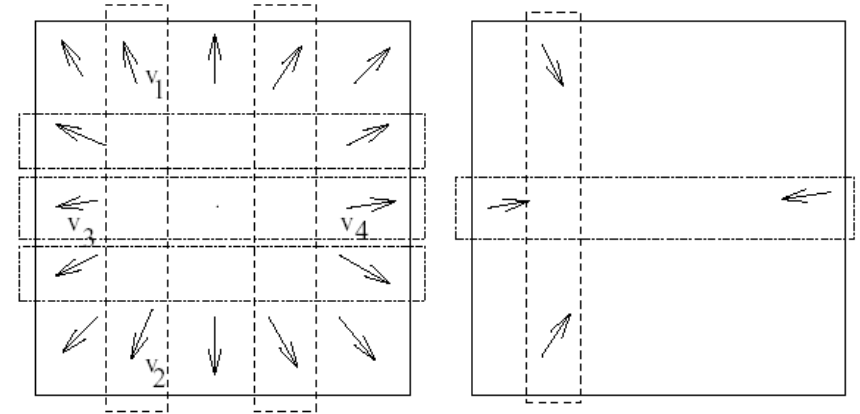
- Wipe



# Como detectar certos efeitos de câmara

- Detecção de zoom

Não é desejável a segmentação do vídeo quando existem alterações significativas na sequência de imagens mas devido a efeitos, como por exemplo, pan ou zoom. Assim, é necessário detectar estes efeitos e fazer merge das subsequências.



Heurística horizontal/vertical

$$|v_{1r} - v_{2r}| > \max\{|v_{1r}|, |v_{2r}|\}$$

$$|v_{3c} - v_{4c}| > \max\{|v_{3c}|, |v_{4c}|\}$$

- Detecção de shots
- Keyframes
- Clips de vídeo; acesso aleatório
- Bases de dados de imagens

Assim que um vídeo longo seja segmentado num conjunto de subsequências, estas subsequências podem ser armazenadas numa base de dados de vídeo com o objectivo de fazer posterior pesquisa.

O vídeo pode ser caracterizando por uma representação hierárquica de cenas e shots, em que cada shot pode ser caracterizado por uma key-frame.

A norma MPEG7 contém um conjunto de descritores e esquemas de descrição que permitem representar ficheiros áudio-visuais (os vídeos são um caso particular) de uma forma normalizada e assim facilitar a pesquisa de informação com base no conteúdo (em alternativa a pesquisa de palavras-chave).

## Content based Image Retirement

<https://arxiv.org/pdf/2012.00641.pdf>

<https://www.tandfonline.com/doi/full/10.1080/23311916.2021.1927469>

[http://printart.isr.ist.utl.pt/my\\_article.pdf](http://printart.isr.ist.utl.pt/my_article.pdf)

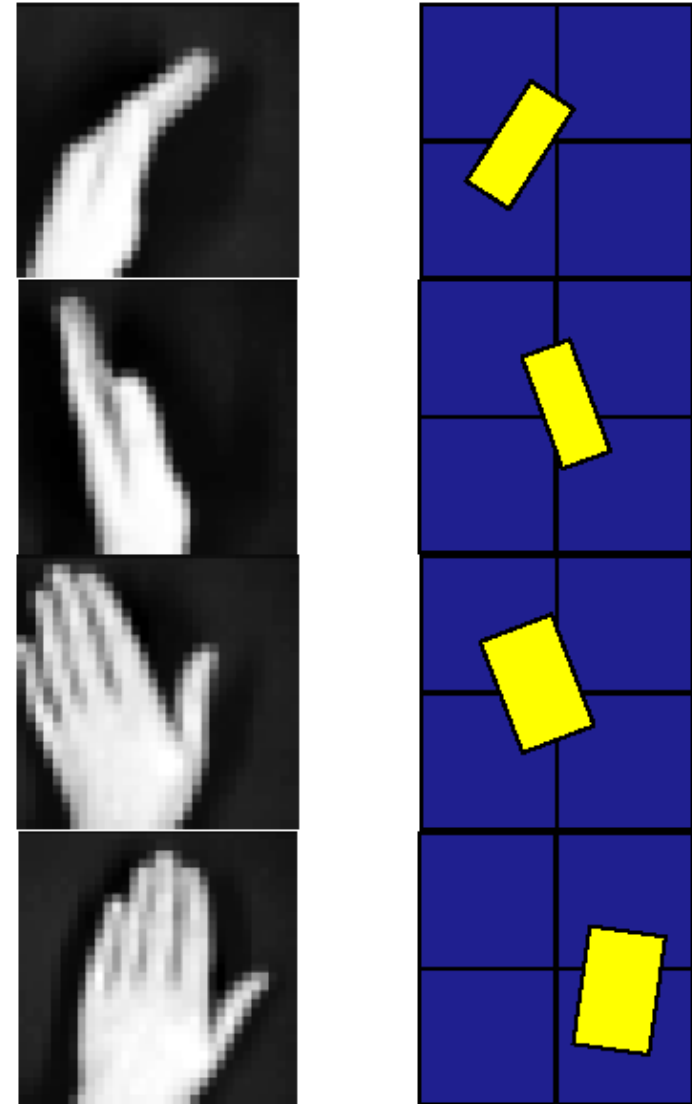
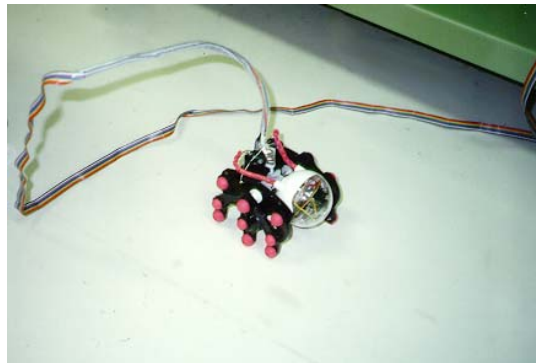
# Computer Vision for Interactive Computer Graphics

---

- Interacção Pessoa – Máquina
- Os computadores são capazes de interpretar movimentos, gestos e olhares do utilizador
- Algoritmos visuais de suporte:
  - Seguimento de objectos;
  - Reconhecimento de formas;
  - Análise de Movimento.
- Em aplicações gráficas interactivas, estes algoritmos necessitam de ser: robustos, rápidos e correr em hardware genérico.
- Felizmente, as aplicações interactivas também facilitam o problema da visão:
  - Restringe o número de interpretações possíveis.
  - Fornece retroacção visual útil

- 
- Aplicações interactivas colocam desafios particulares:
    - O tempo de resposta deverá ser muito rápido
    - Os algoritmos de visão por computador devem ser robustos, funcionarem para diferentes tipos de pessoas e fundos diferenciados.
    - Devem ser baratos. Um joystick actual ou comando de televisão custa 40€ Mesmo com aumento de funcionalidades os consumidores não quererão pagar muito mais.
  - As boas notícias:
    - O contexto da aplicação restringe muito as interpretações visuais possíveis (pelo contexto do jogo sabe-se que a pessoa está a correr; apenas importa definir a velocidade com que a corrida é realizada)
    - Existe um ser inteligente (humano) no ciclo. O utilizador pode explorar a rápida retroacção visual que lhe é fornecida via display gráfico para alterar o seu gesto, se necessário, para alcançar o efeito desejado. Se o jogador se inclina para fazer uma curva e observa que não está a curvar o suficiente, então dever-se-á inclinar mais.

- Existe um nicho para aplicação de algoritmos de visão rápidos e pouco sofisticados que tiram vantagem das simplificações que as aplicações gráficas interactivas introduzem.
- Semelhante em espírito a uma corrente na visão artificial, chamada de visão activa, que enfatiza (tira partido) da resposta em tempo-real dos sistemas reais de visão.



- 
- Visão (uma câmara) pode ser um potente dispositivo de interface em computadores
    - Posição do corpo
    - Orientação da cabeça
    - Direcção do olhar
    - Gestos
  - Aplicações podem incluir:
    - Controlo (via computador) de jogos ou máquinas
    - Interface natural com o computador
  - Em vez de premir botões, o jogador pode realizar mímica de acções e gestos, que o computador reconhece
  - Um utilizador de uma máquina pode dar comandos através de gestos manuais (útil em cirurgias, soldados, deficientes).